M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**BASEXITHA                            C  1




BBBBBBBB      AAAAAA      SSSSSSSS  EEEEEEEEEE  XX      XX    IIIIII    TTTTTTTTTT  HH      HH    AAAAAA
BBBBBBBB      AAAAAA      SSSSSSSS  EEEEEEEEEE  XX      XX    IIIIII    TTTTTTTTTT  HH      HH    AAAAAA
BB      BB  AA      AA  SS          EE          XX      XX      II          TT      HH      HH  AA      AA
BB      BB  AA      AA  SS          EE          XX      XX      II          TT      HH      HH  AA      AA
BB      BB  AA      AA  SS          EE            XX  XX        II          TT      HH      HH  AA      AA
BB      BB  AA      AA  SS          EE            XX  XX        II          TT      HH      HH  AA      AA
BBBBBBBB    AA      AA    SSSSSS    EEEEEEEE        XX          II          TT      HHHHHHHHHH  AA      AA
BBBBBBBB    AA      AA    SSSSSS    EEEEEEEE        XX          II          TT      HHHHHHHHHH  AA      AA
BB      BB  AAAAAAAAAA          SS  EE            XX  XX        II          TT      HH      HH  AAAAAAAAAA
BB      BB  AAAAAAAAAA          SS  EE            XX  XX        II          TT      HH      HH  AAAAAAAAAA
BB      BB  AA      AA          SS  EE          XX      XX      II          TT      HH      HH  AA      AA    ....
BB      BB  AA      AA          SS  EE          XX      XX      II          TT      HH      HH  AA      AA    ....
BBBBBBBB    AA      AA  SSSSSSSS    EEEEEEEEEE  XX      XX    IIIIII        TT      HH      HH  AA      AA    ....
BBBBBBBB    AA      AA  SSSSSSSS    EEEEEEEEEE  XX      XX    IIIIII        TT      HH      HH  AA      AA    ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1            0001  0 MODULE BAS$$EXIT_HANDL (                          ! BASIC exit handler
;     2            0002  0              IDENT = '1-016'                      ! File: BASEXITHA.B32 Edit: PL1016
;     3            0003  0              ) =
;     4            0004  1 BEGIN
;     5            0005  1 !
;     6            0006  1 !****************************************************************************
;     7            0007  1 !*                                                                          *
;     8            0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;     9            0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;    10            0010  1 !*  ALL RIGHTS RESERVED.                                                    *
;    11            0011  1 !*                                                                          *
;    12            0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13            0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;    14            0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15            0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16            0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17            0017  1 !*  TRANSFERRED.                                                            *
;    18            0018  1 !*                                                                          *
;    19            0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20            0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21            0021  1 !*  CORPORATION.                                                            *
;    22            0022  1 !*                                                                          *
;    23            0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24            0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    25            0025  1 !*                                                                          *
;    26            0026  1 !*                                                                          *
;    27            0027  1 !****************************************************************************
;    28            0028  1
;    29            0029  1 !++
;    30            0030  1 ! FACILITY: BASIC support library - Exit handler
;    31            0031  1 !
;    32            0032  1 ! ABSTRACT:
;    33            0033  1 !
;    34            0034  1 !       This module is used when the image exits to do
;    35            0035  1 !       BASIC post processing.  It purges I/O buffers
;    36            0036  1 !       and closes files with proper disposition.
;    37            0037  1 !
;    38            0038  1 ! ENVIRONMENT: User access mode; mixture of AST level or not.
;    39            0039  1 !
;    40            0040  1 ! Author:       John Sauter, Creation date: 23-JAN-1979
;    41            0041  1 !
;    42            0042  1 ! MODIFIED BY:
;    43            0043  1 !
;    44            0044  1 ! 1-001 - Original from FOROPEN.  JBS 23-JAN-1979
;    45            0045  1 ! 1-002 - Call OTS$$PURGE_IOBU to flush any "dirty" buffer.  JBS 24-JAN-1979
;    46            0046  1 ! 1-003 - Move call to OTS$$PURGE_IOBU to OTS$CLOSE_FILE.  JBS 24-JAN-1979
;    47            0047  1 ! 1-004 - Change linkage for OTS$PUSH_CCB to JSB_CB_PUSH and for
;    48            0048  1 !           OTS$POP_CCB to JSB_CB_POP.  JBS 25-JAN-1979
;    49            0049  1 ! 1-005 - Use two dollar signs for non-user entries.  JBS 26-JAN-1979
;    50            0050  1 ! 1-006 - Add OTS$$CLOSE_ALL.  JBS 04-JUN-1979
;    51            0051  1 ! 1-007 - Change to BASIC-specific exit handler.  JBS 16-AUG-1979
;    52            0052  1 ! 1-008 - Call BAS$$PUR_IO_CLO to flush all buffers.  JBS 20-AUG-1979
;    53            0053  1 ! 1-009 - Make BAS$$CLOSE_ALL global, for BAS$$RUN_INIT.  JBS 21-AUG-1979
;    54            0054  1 ! 1-010 - Signal CLOSE errors, but make the severity "warning" so we
;    55            0055  1 !           don't lose control.  JBS 24-AUG-1979
;    56            0056  1 ! 1-011 - Do explicit signalling of CLOSE errors, since OTS$$CLOSE_FILE
;    57            0057  1 !           doesn't.  JBS 27-AUG-1979
```

```
:    58           0058  1 ! 1-012 - Give CLOSE_ALL an optional parameter, so we can close all of the
:    59           0059  1 !            streams connected to a base file.  JBS 28-SEP-1979
:    60           0060  1 ! 1-013 - Clear BAS$$L_XIT_LOCK upon entry to the exit handler.  This
:    61           0061  1 !           allows user exit handlers to perform I/O, and get the proper
:    62           0062  1 !           cleanup upon leaving.
:    63           0063  1 ! 1-014 - If There is a file X that Y and Z have connected to ( via open
:    64           0064  1 !           clause CONNECT) then close Y and Z first and then close X.
:    65           0065  1 !           FM 12-aug-81.
:    66           0066  1 ! 1-015 - LIB$STOP should be declared EXTERNAL.  PLL 20-NOV-81
:    67           0067  1 ! 1-016 - Edit 1-014 breaks virtual files.  BAS$$CLOSE_ALL no longer
:    68           0068  1 !           tried to close them if they were open because LUB$V_M_STR_C
:    69           0069  1 !           was not set.  PLL 24-Feb-82
:    70           0070  1 !--
:    71           0071  1
:    72           0072  1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1
1 REQUIRE 'RTLIN:OTSLNK';                               ! Define all linkages
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$DECL_EXITH : NOVALUE,                         ! Declare EXIT handler
1     EXIT_HANDLER : NOVALUE,                           ! Exit Handler
1     BAS$CLOSE_ALL : NOVALUE,                          ! Close all files
1     TRY_TO_CLOSE : CALL_CCB NOVALUE,                  ! Subroutine for EXIT_HANDLER
1     CLOSE_HANDLER;                                    ! Handler for CLOSE errors
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'RTLML:OTSLUB';                               ! logical Unit Block definitions
1
1 REQUIRE 'RTLIN:OTSMAC';                               ! macros
1
1 REQUIRE 'RTLIN:BASIOERR';                             ! I/O error codes
1
1 REQUIRE 'RTLIN:RTLPSECT';                             ! Define DELCARE_PSECTS macro
1
1 LIBRARY 'RTLSTARLE';                                  ! STARLET library for macros and symbols
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !       NONE
1 !
1 ! PSECT DECLARATIONS:
1 !
1 DECLARE_PSECTS (BAS);                                 ! declare PSECTs for BAS$ facility
1 !
1 ! OWN STORAGE:
1 !
1
1 OWN
1     EXIT_REASON,                                      ! VMS stuffs with reason for exiting
1     EXIT_BLOCK : VECTOR [4] INITIAL (0,
1                                             ! Filled in by VMS with forward link to next EXIT control block
1           0,                                      ! Set to EXIT_HANDLER if RTL sets up EXIT handler
1           0,                                      ! Set to arg count (1) if RTL sets up EXIT handler
1           0);                                     ! Set to EXIT_REASON if RTL sets up EXIT handler
1
1   GLOBAL
1       BAS$$L_XIT_LOCK : INITIAL (0);              ! Clear if no handler linked yet
1
1   !
1   ! (Used to make sure only one handler even if ASTs)
1   !
1   ! EXTERNAL REFERENCES:
1   !
1
1   EXTERNAL LITERAL
1       OTS$_FATINTERR;                             ! OTS Fatal Internal Error
1
1   EXTERNAL ROUTINE
1       LIB$STOP : NOVALUE,                         ! Signal a fatal error
1       BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,         ! Load register CCB
1       BAS$$CB_POP : JSB_CB_POP NOVALUE,           ! Done with register CCB
1       BAS$$NEXT_LUN : NOVALUE,                    ! Get next LUN that might be open
1       BAS$$PUR_IO_CLO : NOVALUE,                  ! Purge all I/O buffers
1       OTS$$CLOSE_FILE : CALL_CCB,                 ! Internal file closer
1       BAS$$SIGNAL_IO : CALL_CCB NOVALUE;          ! Signal a BASIC I/O error
1
```

```
1 GLOBAL ROUTINE BAS$$DECL_EXITH                  ! Declare VMS EXIT handler
1     : NOVALUE =                                 !
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Declares VMS EXIT handler for BASIC.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       IF (NOT .BAS$$L_XIT_LOCK) THEN BAS$$DECL_EXITH ()
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Declares VMS EXIT handler.
1 !--
1
2     BEGIN
2
2     LOCAL
2         AST_STATUS,
2         DCLEXH_STATUS;
2
2 !+
2 ! We must disable ASTs to be sure that one and only one exit handler
2 ! is declared for BASIC.
2 !-
2     AST_STATUS = $SETAST (ENBFLG = 0);
2
3     IF ( NOT .BAS$$L_XIT_LOCK)
2     THEN
3         BEGIN
3 !+
3 ! Initialize EXIT handler control block (must do at run time to be PIC)
3 !-
3         EXIT_BLOCK [1] = EXIT_HANDLER;          ! Adr. of EXIT handler to be called on EXIT
3         EXIT_BLOCK [2] = 1;                     ! arg count
3         EXIT_BLOCK [3] = EXIT_REASON;           ! adr. to store reason for EXIT
3         DCLEXH_STATUS = $DCLEXH (DESBLK = EXIT_BLOCK);
3         BAS$$L_XIT_LOCK = 1;
```

```
;  214          1118 3            END
;  215          1119 2        ELSE
;  216          1120 2            DCLEXH_STATUS = 1;
;  217          1121 2
;  218          1122 2        IF (.AST_STATUS EQL SS$_WASSET) THEN $SETAST (ENBFLG = 1);
;  219          1123 2
;  220          1124 2        IF ( NOT .DCLEXH_STATUS) THEN LIB$STOP (OTS$_FATINTERR);
;  221          1125 2
;  222          1126 2        RETURN
;  223          1127 1        END;


                                                                  .TITLE  BAS$$EXIT_HANDL
                                                                  .IDENT  \1-016\

                                                                  .PSECT  _BAS$DATA,NOEXE,  PIC,2

                                                00000 EXIT_REASON:
                                                                  .BLKB   4
            00000000  00000000  00000000  00000000  00004 EXIT_BLOCK:
                                                                  .LONG   0, 0, 0, 0                          ;
                                      00000000  00014 BAS$$L_XIT_LOCK::
                                                                  .LONG   0                                   ;

                                                                  .EXTRN  OTS$_FATINTERR, LIB$STOP
                                                                  .EXTRN  BAS$$CB_PUSH, BAS$$CB_POP
                                                                  .EXTRN  BAS$$NEXT_LUN, BAS$$PUR_IO_CLO
                                                                  .EXTRN  OTS$$CLOSE_FILE
                                                                  .EXTRN  BAS$$SIGNAL_IO, SYS$SETAST
                                                                  .EXTRN  SYS$DCLEXH

                                                                  .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                                          003C 00000              .ENTRY  BAS$$DECL_EXITH, Save R2,R3,R4,R5   ; 1061
                  55 0000000G   00    9E 00002              MOVAB   SYS$SETAST, R5                      ;
                  54 00000000'  EF    9E 00009              MOVAB   BAS$$L_XIT_LOCK, R4                 ;
                                7E    D4 00010              CLRL    -(SP)                               ; 1105
                  65            01    FB 00012              CALLS   #1, SYS$SETAST                      ;
                  53            50    D0 00015              MOVL    R0, AST_STATUS                      ;
                  21            64    E8 00018              BLBS    BAS$$L_XIT_LOCK, 1$                 ; 1107
        F4        A4     0000V  CF    9E 0001B              MOVAB   EXIT_HANDLER, EXIT_BLOCK+4          ; 1113
        F8        A4            01    D0 00021              MOVL    #1, EXIT_BLOCK+8                    ; 1114
        FC        A4        EC  A4    9E 00025              MOVAB   EXIT_REASON, EXIT_BLOCK+12          ; 1115
                            F0  A4    9F 0002A              PUSHAB  EXIT_BLOCK                          ; 1116
         0000000G 00            01    FB 0002D              CALLS   #1, SYS$DCLEXH                      ;
                  52            50    D0 00034              MOVL    R0, DCLEXH_STATUS                   ;
                  64            01    D0 00037              MOVL    #1, BAS$$L_XIT_LOCK                 ; 1117
                                03    11 0003A              BRB     2$                                  ; 1107
                  52            01    D0 0003C 1$:          MOVL    #1, DCLEXH_STATUS                   ; 1120
                  09            53    D1 0003F 2$:          CMPL    AST_STATUS, #9                      ; 1122
                                05    12 00042              BNEQ    3$                                  ;
                                01    DD 00044              PUSHL   #1                                  ;
                  65            01    FB 00046              CALLS   #1, SYS$SETAST                      ;
                  0D            52    E8 00049 3$:          BLBS    DCLEXH_STATUS, 4$                   ; 1124
                       0000000G 8F    DD 0004C              PUSHL   #OTS$_FATINTERR                     ;
         0000000G 00            01    FB 00052              CALLS   #1, LIB$STOP                        ;
                                      04 00059 4$:          RET                                         ; 1127
```

```
; Routine Size:  90 bytes,    Routine Base:  _BAS$CODE + 0000

;  224          1128  1
;  226          1129  1 ROUTINE EXIT_HANDLER (                                  ! Exit Handler
;  227          1130  1         EXIT_REASON                                     ! Reason
;  228          1131  1     ) : NOVALUE =
;  229          1132  1
;  230          1133  1 !++
;  231          1134  1 ! FUNCTIONAL DESCRIPTION:
;  232          1135  1 !
;  233          1136  1 !       This is the exit handler for BASIC.  Its only function is to
;  234          1137  1 !       purge I/O buffers and close all files.
;  235          1138  1 !
;  236          1139  1 !       Upon entry, it zeroes BAS$$L_XIT_LOCK so that user I/O in
;  237          1140  1 !       exit handlers can get properly cleaned up.
;  238          1141  1 !
;  239          1142  1 ! FORMAL PARAMETERS:
;  240          1143  1 !
;  241          1144  1 !       EXIT_REASON.rl.r        not used
;  242          1145  1 !
;  243          1146  1 ! IMPLICIT INPUTS:
;  244          1147  1 !
;  245          1148  1 !       NONE
;  246          1149  1 !
;  247          1150  1 ! IMPLICIT OUTPUTS:
;  248          1151  1 !
;  249          1152  1 !       BAS$$L_XIT_LOCK is zeroed.
;  250          1153  1 !
;  251          1154  1 ! ROUTINE VALUE:
;  252          1155  1 ! COMPLETION CODES:
;  253          1156  1 !
;  254          1157  1 !       NONE
;  255          1158  1 !
;  256          1159  1 ! SIDE EFFECTS:
;  257          1160  1 !
;  258          1161  1 !       Closes all files by calling BAS$$CLOSE_ALL.
;  259          1162  1 !--
;  260          1163  1
;  261          1164  2     BEGIN
;  262          1165  2     BAS$$L_XIT_LOCK = 0;                                ! Clear exit handler interlock
;  263          1166  2     BAS$$CLOSE_ALL ();
;  264          1167  1     END;                                                ! of routine EXIT_HANDLER



                                          0000 00000 EXIT_HANDLER:
                                                          .WORD    Save nothing                               ; 1129
                           00000000'  EF  D4 00002        CLRL     BAS$$L_XIT_LOCK                            ; 1165
                   0000V  CF          00  FB 00008        CALLS    #0, BAS$$CLOSE_ALL                         ; 1166
                                          04 0000D        RET                                                 ; 1167

; Routine Size:  14 bytes,    Routine Base:  _BAS$CODE + 005A


;  265          1168  1
```

```
1 GLOBAL ROUTINE BAS$$CLOSE_ALL (                        ! Close all files
1       PARENT_IFI                                      ! Optional IFI to look for
1    ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Find every existing LUB (with a linear search through the LUB
1 !       table).  For each LUB, if the file is open, purge its I/O
1 !       buffers and close it.  If the file has been marked for PRINT
1 !       or DELETE, this will cause proper disposition of the file.
1 !       RMS will close all open files at image exit, but it doesn't know
1 !       about the above two DISPOSE conditions.  We couldn't set them at
1 !       OPEN time, since the user is allowed to specify a different
1 !       DISPOSE option at close time (with the CLOSE statement).
1 !       Note that BASIC does not yet have CLOSE options, so this code is
1 !       a provision for the future.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       PARENT_IFI.rl.v If present, close all files with M_STREAM set
1 !                       and this IFI.  This is used by CLOSE when closing
1 !                       a file which has multiple streams.  The calls to
1 !                       OTS$$CLOSE_FILE will actually result in $DISCONNECTs.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Closes all files.
1 !       Signals CLOSE and DISCONNECT errors as warnings.
1 !--
1
2    BEGIN
2
2    BUILTIN
2       NULLPARAMETER;
2
2    GLOBAL REGISTER
2       CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2    LOCAL
2       FLAG,
2       LUN;
2
2 !+
2 ! Scan through all BASIC logical units, closing them.
2 !-
2      FLAG = 0;
2
2      DO
3          BEGIN
3 !+
3 ! Get the next logical unit number.
3 !-
3          BAS$$NEXT_LUN (FLAG, LUN);
3
4          IF (.FLAG NEQ 0)
3          THEN
4              BEGIN
4 !+
4 ! LUN is the next logical unit number.  If the file it represents is
4 ! open try to close it.
4 !-
4              BAS$$CB_PUSH (.LUN, LUB$K_ILUN_MIN);
4
5              IF (.CCB [LUB$V_OPENED])
4              THEN
5                  BEGIN
5
6                  IF (NULLPARAMETER (1))
5                  THEN
6                      BEGIN
6
7                      IF (.CCB [LUB$V_M_STR_C])
6                      THEN
7                          BEGIN
7 !+
7 ! Close all the sons of the mother before closing the mother, i.e. if Y and
7 ! Z are connected to X, and we are closing X, then we must close Y and Z and
7 ! then close X.
7 !-
7                          BAS$$CLOSE_ALL (.CCB [LUB$W_IFI]);
7                          TRY_TO_CLOSE ();
7                          END
6                      ELSE
6                          TRY_TO_CLOSE ();
6
6                      END
5                  ELSE
5 !+
5 ! Do the close (actually disconnect) only if
5 ! the IFI matches and this is a connect.
5 !-
5
5                      IF (.CCB [LUB$V_M_STREAM] AND (.CCB [LUB$W_IFI] EQL .PARENT_IFI)) THEN TRY_TO_CLOSE ();
4                  END;
4
4              BAS$$CB_POP ();
3              END;
3
3          END
2      UNTIL (.FLAG EQL 0);
```

```
;  381          1283  2
;  382          1284  2      RETURN;
;  383          1285  1      END;                                          ! of routine BAS$$CLOSE_ALL



                                                        0804 00000          .ENTRY   BAS$$CLOSE_ALL, Save R2,R11                  ; 1169
                                        5E             08 C2 00002          SUBL2    #8, SP                                       ;
                                                   04  AE D4 00005          CLRL     FLAG                                         ; 1228
                                                       5E DD 00008 1$:      PUSHL    SP                                           ; 1235
                                                   08  AE 9F 0000A          PUSHAB   FLAG                                         ;
                            00000000G   00             02 FB 0000D          CALLS    #2, BAS$$NEXT_LUN                            ;
                                                   04  AE D5 00014          TSTL     FLAG                                         ; 1237
                                                       41 13 00017          BEQL     6$                                           ;
                                        50             08 CE 00019          MNEGL    #8, R0                                       ; 1244
                                        52             6E D0 0001C          MOVL     LUN, R2                                      ;
                                            00000000G  00 16 0001F          JSB      BAS$$CB_PUSH                                 ;
                                        2B         FC  AB E9 00025          BLBC     -4(CCB), 5$                                  ; 1246
                                                       6C 95 00029          TSTB     (AP)                                         ; 1250
                                                       05 13 0002B          BEQL     2$                                           ;
                                                   04  AC D5 0002D          TSTL     4(AP)                                        ;
                                                       0F 12 00030          BNEQ     3$                                           ;
                       18         FF    AB             03 E1 00032 2$:      BBC      #3, -1(CCB), 4$                              ; 1254
                                        7E         D0  AB 3C 00037          MOVZWL   -48(CCB), -(SP)                              ; 1262
                                   C1   AF             01 FB 0003B          CALLS    #1, BAS$$CLOSE_ALL                           ;
                                                       0E 11 0003F          BRB      4$                                           ; 1266
                       0E         FF    AB             02 E1 00041 3$:      BBC      #2, -1(CCB), 5$                              ; 1275
    04   AC       D0   AB               10             00 ED 00046          CMPZV    #0, #16, -48(CCB), PARENT_IFI                ;
                                                       05 12 0004D          BNEQ     5$                                           ;
                                 0000V  CF             00 FB 0004F 4$:      CALLS    #0, TRY_TO_CLOSE                             ;
                                            00000000G  00 16 00054 5$:      JSB      BAS$$CB_POP                                  ; 1278
                                                   04  AE D5 0005A 6$:      TSTL     FLAG                                         ; 1282
                                                       A9 12 0005D          BNEQ     1$                                           ;
                                                          04 0005F          RET                                                   ; 1285

; Routine Size:  96 bytes,    Routine Base:  _BAS$CODE + 0068

;  384          1286  1
;  386          1287  1 ROUTINE TRY_TO_CLOSE                                         ! Call OTS$$CLOSE_FILE with errors as warnings
;  387          1288  1     : CALL_CCB NOVALUE =
;  388          1289  1
;  389          1290  1 !++
;  390          1291  1 ! FUNCTIONAL DESCRIPTION:
;  391          1292  1 !
;  392          1293  1 !       RMS CLOSE a file (by calling OTS$$CLOSE_FILE) but signal errors as warnings, to
;  393          1294  1 !       avoid losing control.
;  394          1295  1 !
;  395          1296  1 ! FORMAL PARAMETERS:
;  396          1297  1 !
;  397          1298  1 !       NONE
;  398          1299  1 !
;  399          1300  1 ! IMPLICIT INPUTS:
;  400          1301  1 !
;  401          1302  1 !       CCB             Pointer to the LUB/ISB/RAB of the file to CLOSE.
;  402          1303  1 !
;  403          1304  1 ! IMPLICIT OUTPUTS:
;  404          1305  1 !
;  405          1306  1 !       NONE
;  406          1307  1 !
;  407          1308  1 ! ROUTINE VALUE:
;  408          1309  1 ! COMPLETION CODES:
;  409          1310  1 !
;  410          1311  1 !       NONE
;  411          1312  1 !
;  412          1313  1 ! SIDE EFFECTS:
;  413          1314  1 !
;  414          1315  1 !       RMS CLOSEs the file.
;  415          1316  1 !       Signals CLOSE errors as warnings.
;  416          1317  1 !--
;  417          1318  1
;  418          1319  2     BEGIN
;  419          1320  2
;  420          1321  2     EXTERNAL REGISTER
;  421          1322  2         CCB : REF BLOCK [, BYTE];
;  422          1323  2
;  423          1324  2     ENABLE
;  424          1325  2         CLOSE_HANDLER ();
;  425          1326  2
;  426          1327  2 !+
;  427          1328  2 ! Write output buffers, then RMS CLOSE the file.
;  428          1329  2 !-
;  429          1330  2     BAS$$PUR_IO_CLO ();
;  430          1331  2
;  431          1332  2     IF ( NOT OTS$$CLOSE_FILE ()) THEN BAS$$SIGNAL_IO (BAS$K_IOERR_REC);
;  432          1333  2
;  433          1334  2     RETURN;
;  434          1335  1     END;



                                                   0000 00000 TRY_TO_CLOSE:
                                                                   .WORD   Save nothing                                    ; 1287
                                    6D      001D   CF   DE 00002   MOVAL   2$, (FP)                                        ; 1319
```

```
                    00000000G  00           00  FB 00007          CALLS   #0, BAS$$PUR_IO_CLO                    ; 1330
                    00000000G  00           00  FB 0000E          CALLS   #0, OTS$$CLOSE_FILE                    ; 1332
                               0A           50  E8 00015          BLBS    R0, 1$                                 ;
                               7E           01  CE 00018          MNEGL   #1, -(SP)                              ;
                    00000000G  00           01  FB 0001B          CALLS   #1, BAS$$SIGNAL_IO                     ;
                                                04 00022  1$:     RET                                            ; 1335
                                              0000 00023  2$:     .WORD   Save nothing                           ; 1319
                                            7E  D4 00025          CLRL    -(SP)                                  ;
                                            5E  DD 00027          PUSHL   SP                                     ;
                               7E      04   AC  7D 00029          MOVQ    4(AP), -(SP)                           ;
                        0000V  CF           03  FB 0002D          CALLS   #3, CLOSE_HANDLER                      ;
                                                04 00032          RET                                            ;

; Routine Size:  51 bytes,    Routine Base:  _BAS$CODE + 00C8
```

```
1 ROUTINE CLOSE_HANDLER (                                   ! Handle an error from CLOSE_ALL
1         SIG,                                              ! Signal vector
1         MECH,                                             ! Mechanism vector
1         ENBL                                              ! Enable vector
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       If we get an error trying to close a file in CLOSE_ALL, convert the severity
1 !       of the error to WARNING (if it is ERROR or SEVERE ERROR) so that we will not
1 !       lose control.  It is important not to lose control so that we can try (at least)
1 !       to close all the files.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       SIG.rl.a        A counted vector of parameters to LIB$SIGNAL/STOP
1 !       MECH.rl.a       A counted vector of info from CHF
1 !       ENBL.ra.a       A counted vector of ENABLE argument addresses.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION CODES:
1 !
1 !       Always SS$_RESIGNAL, which is ignored when unwinding.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Reduces the severity of the error to WARNING.
1 !
1 !--
1
2     BEGIN
2
2     MAP
2         SIG : REF VECTOR,
2         MECH : REF VECTOR,
2         ENBL : REF VECTOR;
2
2     LOCAL
2         COND_VALUE : BLOCK [4, BYTE];
2
2     COND_VALUE = .SIG [1];
2 !+
2 ! If the severity is ERROR or SEVERE ERROR, convert it to WARNING.
2 !-
2
2     SELECTONE .COND_VALUE [STS$V_SEVERITY] OF
2         SET
2
2         [STS$K_ERROR, STS$K_SEVERE] :
```

```
;  493            1393  3                 BEGIN
;  494            1394  3                 COND_VALUE [STS$V_SEVERITY] = STS$K_WARNING;
;  495            1395  3                 SIG [1] = .COND_VALUE;
;  496            1396  2                 END;
;  497            1397  2
;  498            1398  2             [OTHERWISE] :
;  499            1399  3                 BEGIN
;  500            1400  3                 0
;  501            1401  2                 END;
;  502            1402  2             TES;
;  503            1403  2
;  504            1404  2         RETURN (SS$_RESIGNAL);
;  505            1405  1         END;                                          ! end of CLOSE_HANDLER



                                                      0000 00000 CLOSE_HANDLER:
                                                                         .WORD    Save nothing                                  ; 1336
                                 50        04    AC   D0 00002           MOVL     SIG, R0                                       ; 1384
                                 51        04    A0   D0 00006           MOVL     4(R0), COND_VALUE                             ;
       02             51         03              00   ED 0000A           CMPZV    #0, #3, COND_VALUE, #2                        ; 1392
                                                 07   13 0000F           BEQL     1$                                            ;
       04             51         03              00   ED 00011           CMPZV    #0, #3, COND_VALUE, #4                        ;
                                                 07   12 00016           BNEQ     2$                                            ;
                                 51              07   8A 00018 1$:       BICB2    #7, COND_VALUE                                ; 1394
                            04   A0              51   D0 0001B           MOVL     COND_VALUE, 4(R0)                             ; 1395
                                 50      0918    8F   3C 0001F 2$:       MOVZWL   #2328, R0                                     ; 1404
                                                      04 00024           RET                                                    ; 1405

; Routine Size:  37 bytes,    Routine Base:  _BAS$CODE + 00FB


;  506            1406  1 END                                           ! End of BAS$$EXIT_HANDL module
;  507            1407  1
;  508            1408  0 ELUDOM





;                              PSECT SUMMARY
;
;       Name                Bytes                        Attributes
;
;  _BAS$DATA                        24  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _BAS$CODE                       288  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)




;                         Library Statistics
;
;                                    -------- Symbols --------       Pages        Processing
;       File                         Total     Loaded   Percent      Mapped       Time
```

```
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1          9776        10          0       581           00:01.2





;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASEXITHA/OBJ=OBJ$:BASEXITHA MSRC$:BASEXITHA/UPDATE=(ENH$:BASEXITHA
;       )

; Size:          288 code + 24 data bytes
; Run Time:         00:12.0
; Elapsed Time:     00:28.2
; Lines/CPU Min:     7028
; Lexemes/CPU-Min: 31202
; Memory Used:  115 pages
; Compilation Complete
```

0023 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASFREE
LIS

BASEXITHA
LIS

BASFETCHD
LIS

BASFORINT
LIS

BASGETRFA
LIS

BASFETCHA
LIS

BASGET
LIS

BASFSP
LIS

BASFIND
LIS

BASFORMAT
LIS

BASHANDLE
LIS